```sql
-- Project name: Starlink-Inspired Telemetry Analysis

-- Dataset used: "fraudTrain.csv" from
--  the Credit Card Transactions Fraud Detection Dataset by Kartik Shenoy,
--  who generated the data using the Sparkov Data Generation | Github tool by Brandon Harris
--  (source: Kaggle, https://www.kaggle.com/datasets/kartik2112/fraud-detection/data?select=fraudTrain.csv ).

-- Project by Ethan Troy Sanchez

-- EXPLORATORY SECTION:

SELECT COUNT(*) AS Total
FROM transactions;
-- The output is Total = 1296675 .

SELECT is_fraud, COUNT(*) AS FraudCount
FROM transactions
GROUP BY is_fraud; -- '0' is legitimate, '1' is fraud
-- The output is
--  is_fraud == 0, FraudCount = 1289169
--  is_fraud == 1, FraudCount = 7506 .

SELECT ( COUNT(*)
  FILTER (WHERE is_fraud = 0) * 1.0 / COUNT(*) ) * 100 AS SuccessRate
FROM transactions;
-- The output is SuccessRate = 99.4211348256117 ,
-- a 99.42 percent success rate.

SELECT ( COUNT(*)
  FILTER (WHERE is_fraud = 1) * 1.0 / COUNT(*) ) * 100 AS FraudRate
FROM transactions;
-- The output is FraudRate = 0.5788651743883395 ,
-- a 0.58 percent fraudulent rate

-- This is the baseline fraud rate. High conversion, low fraud:
-- a 0.58% fraud rate is equivalent to 7506 fraudulent transactions (txns).

-- ANALYSIS SECTION:

-- Metric #1: success rate by category
SELECT category,
  COUNT(*) AS Txns,
  SUM( CASE
    WHEN is_fraud = 0 THEN 1
    ELSE 0
    END ) AS Legit,
  ( SUM( CASE
```

```
48        WHEN is_fraud = 0 THEN 1
49        ELSE 0
50        END ) * 1.0 / COUNT(*) ) * 100 AS SuccessRateByCategory
51  FROM transactions
52  GROUP BY category
53  ORDER BY SuccessRateByCategory;
54  -- category mimics how Starlink's payment methods vary.
55  -- For category, shopping_net is 98.2% successful,
56  --                   misc_net is 98.55% successful,
57  --                   grocery_pos is 98.59% successful, etc.
58  -- Starlink's payment methods vary too, like credit, PayPal, etc.
59  -- The lower success rates are the optimization targets, the areas to pay attention to.
60  -- The output is saved as "success_by_category.csv".
61
62  -- Metric #2: fraud costs by state
63  SELECT state,
64    SUM( CASE
65        WHEN is_fraud = 1 THEN amt
66        ELSE 0
67        END ) AS FraudCost,
68    COUNT( CASE
69        WHEN is_fraud = 1 THEN 1
70        END ) AS FraudCount
71  FROM transactions
72  GROUP BY state
73  ORDER BY FraudCost DESC;
74  -- This query (especially by ordering in descending order)
75  --  highlights where the regional fraud spikes are,
76  --  which, in the case of Starlink, could tank its costs
77  --  if responded to accordingly.
78  -- This would be similar to handling multi-processor issues.
79  -- This query emphasizes the fraud cost per state
80  --  to mimic processor losses.
81  -- For example, with this descending order via FraudCost,
82  --  the output shows that the top 5 states with the highest fraud counts
83  --  and fraud costs are:
84  --   New York (NY), with 555 frauds totaling to $295,548.64
85  --   Texas (TX), with 479 frauds totaling to $265,806.41
86  --   Pennsylvania (PA), with 458 frauds totaling to $244,624.67
87  --   California (CA), with 326 frauds totaling to $170,943.92
88  --   Ohio (OH), with 321 frauds totaling to $168,919.98 .
89  -- For SpaceX, these would constitute Starlink wins,
90  --  as cutting fraud in significant high-cost states like CA and TX with tighter checks
91  --  would likely increase the future success rates set up in Metric #1.
92  -- The output is saved as "fraud_by_state.csv".
93
94  -- Metric #3: Average Amount by Outcome
95  SELECT is_fraud,
```

```
    AVG(amt) AS AvgAmount,
    COUNT(*) AS Txns
FROM transactions
GROUP BY is_fraud;
-- This query shows that fraud transactions, on average, are more expensive,
--  providing a clue for where to implement cost reduction.
-- This kind of query could be written for a similar situation in SpaceX.
-- The output is saved as "amount_by_outcome.csv".

-- Metric #4: Time Trend
SELECT strftime('%Y-%m-%d', trans_date_trans_time) AS Date,
    COUNT(*) AS Txns,
    SUM(is_fraud) AS Frauds
FROM transactions
GROUP BY Date;
-- This query records the number of transactions and number of frauds
--  per day, from January 1, 2019, to June 21, 2020.
-- This output, saved as "time_trend.csv", can be visualized
--  with time-series data, dashboards, or some other type of sequential tool
--  to show change and potential patterns over time.

```